Formal Goal-based Modeling of Organizations

Viara Popova[1,2], Alexei Sharpanskykh[2]

[1] De Montfort University, Centre for Manufacturing, The Gateway, Leicester, LE1 9BH, UK
vpopova@dmu.ac.uk
[2] Vrije Universiteit Amsterdam, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
sharp@cs.vu.nl

Abstract. Each organization exists for the achievement of certain goals. To ensure continued success, the organization should monitor its performance w.r.t. these goals. Performance is often evaluated by estimating performance indicators (PIs). In most existing organization modeling approaches the relation between PIs and goals is implicit. This paper proposes a formal approach for modeling goals based on PIs and defines mechanisms for establishing goal satisfaction, which enable evaluation of organizational performance. Analysis and methodological issues related to goals are briefly discussed.

1 Introduction

The behavior of an organization is usually guided by its strategic and tactical goals. The performance of an organization is often evaluated by estimating the values of its qualitative and quantitative performance indicators (e.g., profits, number of clients). To ensure effectiveness of an organization, the key PIs should be reflected in its goals. In most existing approaches on organization modeling the relation between PIs and goals remains implicit. This paper defines a clear and general mechanism for specifying goals based on PIs. The organization's performance can then be evaluated by estimating the satisfaction of its goals. Moreover, often the satisfaction of goals can only be established in a framework, in which goals are related to other concepts (such as tasks, roles and agents). A goal-based modeling approach proposed in this paper constitutes a part of a general organization modeling and analysis framework, in which organizations are considered from four interrelated perspectives (or views). In particular, *the performance-oriented view* addresses PIs, goals and relations between them; *the process-oriented view* considers tasks, workflows and resources; the *organization-oriented view* defines roles, their authority and interaction relations; *the agent-oriented view* identifies different types of agents with their capabilities, and principles of allocation of agents to roles. The views are linked via relations between their concepts, e.g., agents are allocated to roles, the roles are assigned tasks, tasks realize goals, etc. Concepts and relations within every view are formally described using dedicated languages expressive enough to convey structures and processes of organizations of most types. To provide the formal meaning for the concepts and to ensure consistency of specifications, axioms and constraints are defined that establish relations between concepts within and across views.

The formal language, axioms and constraints specific for *modeling goals* within the *performance-oriented view* are described in this paper. Moreover, some of the verification techniques specific for the performance-oriented view are briefly considered. Furthermore, the paper addresses methodological issues of creating and revising goal structures, and considers the process of organizational performance evaluation based on goals.

The proposed approach was applied for modeling and analyzing an organization from the security domain within the project Cybernetic Incident Management. The main purpose of the organization is to deliver security services. It has multi-level structure, with predefined job descriptions for employees. The examples given in this paper are related to a part of the organization concerned with planning the allocation of security officers to locations of customers. The planning process consists of forward (or long-term) planning and short-term planning. Forward planning is a process of creation, analysis and optimization of forward plans for the allocation of security officers for a long term. During the short-term planning, plans for the allocation of security officers to locations within a certain area for a short term are created and updated based on the forward plan and up-to-date information about the security employees. Due to the space limitations only some parts of this case study are given in the paper. A more elaborated description of the case study is given in [7].

The paper is organized as follows. In Section 2 the main concepts for the goal modeling framework are specified. The relationships between them are described and formalized using the dedicated logic-based language in Section 3. Section 4 discusses how the performance of the organization can be evaluated. Some design principles are given in Section 5. Section 6 discusses the related work on goal-oriented modeling.

2 Goal Modeling Concepts

Each organization exists for the achievement of one or more goals. This varies depending on the type of organization, e.g., the main goal of a manufacturing company can be the realization of maximal profit. Being aware of these goals is a prerequisite to taking measures for their satisfaction. To ensure continued success, the organization should monitor its performance w.r.t. formulated goals. To enable goal-based performance evaluation, organizational goals should be formulated over performance measures (indicators).

Performance indicators (PIs) are defined as measures, quantitative or qualitative, that can be used to give a view on the state or progress of the company, a unit within the company or an individual (e.g., time to produce a short-term plan, efficiency of allocation of security officers). Expressions can be formulated over PIs containing >, = or <, for example for defining target values: an expression over the PI P1:"efficiency of allocation of security officers" is defined as P1 = high.

Goal pattern is a property over one or more PI expressions used to define a goal. A goal pattern can be checked for a given state/time point or interval for a company or agent. Goal patterns have a *type* specifying the way the property will be checked: *achieved (ceased)* – check if the property is true for a given time point; *maintained* (*avoided*) – check if the property is true for the duration of a given time interval;

optimized (maximized/ minimized/approximated) – check if the value of the PI expression has increased/decreased/approached a target value for the given time interval. *Achieved, ceased, maintained*, *avoided* are used on PI expressions that are evaluated to Boolean values; *optimized* is used on PI expressions that are evaluated to value of any ordered type (*maximized/minimized*) or for which a distance measure is defined (*approximated*). Examples: "maintained efficiency of allocation of security officers = high" with *maintained* pattern type and "achieved that time to produce short-term plan ≤ 48" with *achieved* pattern type. Goals are formulated by adding to goal patterns information such as desirability and priority.

Goal is an objective describing a desired state or development of the company or an individual. Example: "it is required to maintain high efficiency of allocation of security officers". A goal is characterized by *evaluation type, horizon, ownership, hardness*, *priority, negotiability, etc*. *Evaluation type* determines if a goal is based on goal pattern of type *achieved* or *ceased* (*achievement goal*), or if a goal is based on goal pattern *maintained, avoided* or *optimized* (*development goal*). *Horizon* specifies within which time interval (for development goals) or at which time point (for achievement goals) is the goal supposed to be satisfied: *long-, mid-long-* or *short-term goal*. *Ownership* can be *organizational* (of an organization/unit/role) or *individual* (of an agent). Goals of agents may comply with organizational goals or not. *Priority* is defined by the numerical estimation between 0 and 1; alternatively {very high, high, medium, low, very low} or a (partial) ordering on goals may be defined. Normally, organizational goals have a higher level of priority. Priority of individual goals depends on the company policy: one company might assign lower priority to individual goals; another might decide to involve and motivate the agents by taking into account their goals and avoiding conflicts between individual and organizational goals. By *negotiability* goals are divided into *non-negotiable* and *negotiable*. This can be used for conflict resolution at the design phase.

Hardness distinguishes *soft* and *hard* goals. Satisfaction of a soft goal cannot be clearly established. We use the term *satisficing* to indicate acceptable degree of satisfaction of soft goals. Labels are given corresponding to the degrees of satisficing/denial with a natural order: satisficed > weakly_satisficed > undetermined > weakly_denied > denied. Satisfaction of hard goals can be established quantitatively. Hard goals also have labels ordered as follows: satisfied > undetermined > failed. In the example below G3 is soft, PI "efficiency of allocation" cannot be objectively established to be maintained high or not, instead we use a subjective estimation of degree of satisficing. G3.1 is hard – it can be checked if PI "time to produce up-to-date plan" is at most 48 hours.

Goals are realizable by organizational tasks. A *task* represents a function performed by role(s). A *role* is characterized by a set of functionalities performed by it. Roles are allocated to *agents* to perform tasks. Roles and agents are committed to certain goals. Besides organizational goals, an agent may pursue its own individual goals that comply or conflict with organizational goals. These and other concepts are only briefly discussed in this paper and are considered in the descriptions of other views.

<u>*Examples:*</u>

Goal name: G3
Informal definition: It is required to maintain high efficiency of allocation of security officers to objects
Eval. type: development goal (maintain goal pattern)
Horizon: long-term; *Ownership*: organizational
Hardness: soft
Priority: high; *Negotiability*: negotiable

Goal name: G3.1
Informal definition: It is required to achieve that within 48 hours from receiving operational data, an up-to-date short-term plan exists
Eval. type: achievement goal (achieve goal pattern)
Horizon: short-term; *Ownership*: organizational
Hardness: hard
Priority: medium; *Negotiability*: negotiable

3 Formal Goal Modeling

To specify the meta-model for the performance-oriented view the first order sorted predicate language is used (the graphical representation of the developed meta-model is given in [7]). In this language, for each concept a special sort is introduced, which contains all the names of concept instances (e.g., sort GOAL contains all names of goals). Using this dedicated language a number of relations between goals and other concepts are specified. To provide the formal meaning for the introduced relations and to ensure the consistency and integrity of goal-based specifications, axioms and constraints are defined along the definitions of relations.

The relations on goals and PIs introduced informally in Section 2 are formalized as follows:

is_based_on: GOAL_PATTERN × PI: the goal pattern in the first argument is defined over the PI in the second argument; ***uses***: GOAL_PATTERN × PI_EXPRESSION: goal pattern defined over PI expression. For example, the goal pattern GP1 "maintained efficiency of allocation of security officers to objects = high" is based on the PI P2 "efficiency of allocation of security officers to objects" and uses the PI expression PE1 formulated over P2, (PE1: P2=high): is_based_on(GP1,P2); uses(GP1,PE1).

is_formulated_over: GOAL × GOAL_PATTERN: The goal is defined over the goal pattern. For example, goal G3 defined earlier is formulated over the goal pattern GP1.

Goals are related to tasks, roles and agents by the following relations: ***is_realizable_by***: GOAL × TASK_LIST: the goal in the first argument is realizable by the list of tasks in the second argument; ***is_committed_to***: ROLE × GOAL: the goal is an organizational goal and the role is committed to the satisfaction of this goal; ***wishes***: AGENT × GOAL: the goal is an individual goal of the agent. For example role Planner is committed to goal G3.1 realizable by task T4.4.1:"update short-term plan", i.e. is_committed_to(Planner,G3.1) & is_realizable_by(G3.1,L41) & is_in_ task_list(L41,T4.4.1), where is_in_task_list: TASK_LIST × TASK.

A goal structure can be built by refining high level goals and by aggregating lower lever goals into higher level goals. Since goals in the modeling framework can be of two types: hard and soft, having very different features, different types of refinement relations should be considered.

Hard goals are refined into *and-lists* of hard goals (sort AND_GOAL_LIST), in which the goals are connected by *and* relation. A refinement of a hard goal is specified by:

is_refined_to: GOAL × AND_GOAL_LIST: Defines a refinement of a hard goal into a list of hard goals, which contribute to its satisfaction. When all the goals in the list are satisfied then the goal in the first argument is satisfied as well. If one or more goals in the list fail and no other refinement exists where all goals are satisfied, then the goal in the first argument will fail too. More formally, the predicates satisfied: GOAL and failed: GOAL express the satisfaction state of a goal and the following axioms are formulated:

$$\text{is\_refined\_to}(g, l) \;\&\; (\forall gi{:}\text{GOAL is\_in\_ goal\_list}(gi,l) \Rightarrow \text{satisfied}(gi)) \Rightarrow \text{satisfied}(g)$$

$$\forall\, l{:}\ \text{AND\_GOAL\_LIST}\ (\text{is\_refined\_to}(g, l) \Rightarrow \exists gi{:}\ \text{GOAL is\_in\_ goal\_list}(gi, l) \;\&\; \text{failed}(gi)) \Rightarrow \text{failed}(g)$$

where ***is_in_goal_list***: GOAL × GOAL_LIST expresses that a goal is in a goal list. Sort GOAL_LIST is a supersort of AND_GOAL_LIST, which contains the names of all goal lists.

When more than one refinements are defined, they are considered as alternatives connected by OR, i.e., they allow a choice, which measures to take to satisfy the goal. Examples of refinement of hard goals are provided in [7].

Since the satisfaction of ***soft goals*** cannot be established in a clear-cut way, their refinement differs from the refinement of hard goals. Instead of decomposition, we talk about positive contribution from other goals to the satisfaction of the goal. Such contribution can vary in its degree expressed by the following relations, in which the second argument is a soft goal and the first argument can be soft or hard. In ***satisfices***: GOAL × GOAL the first goal strongly contributes positively to the satisficing of the second goal. If it is satisfi(c)ed and no other influences are known then the second goal is considered satisficed. In ***contributes_to***: GOAL × GOAL the first goal contributes positively to the satisficing of the second goal, but might not be enough to satisfice it.

The precise meaning of these relations is defined through the propagation rules for goals refinement. These rules are used to determine the degree of satisficing of a higher level goal (specified by a label) based on the available information about the degrees of satisfaction/satisficing of lower level goals in its refinement list. To determine the label of a higher level goal, first the labels of its contributing (lower level) goals are propagated, taking into account the types of the links by which the lower level goals are connected to the higher level goal. The propagated labels of the lower level goals are determined using Table 1. Then, the propagated labels of the lower level goals are combined depending on the type of the list to determine the label of the higher level goal.

Lower level goals can be combined in lists using *and-* or *balanced contribution* relations, contributing positively to the satisficing of higher level soft goals:
has_influence_from: GOAL × GOAL_LIST: The goals in the list contribute positively to the satisficing of the soft goal in the first argument. For each goal in the list it is defined separately what the level is of its contribution (the type of the link) using the above defined relations satisfices and contributes_to.

The combination of goals in an and-list implies that if all goals in the list are satisfi(c)ed then the higher level goal will also be satisficed. In order to ensure this the following constraint is enforced: at least one of the goals in an and-list is connected with a link of the type satisfices to the higher level goal. When lower level goals are combined in an and-list, the label of a higher level goal is defined by the minimal label propagated from the goals in this list using the defined order between the labels.

Another kind of relation between goals represents balanced contribution which gives us the possibility to describe more fine-tuned ways of contributing which favor the majority influence. The rule that is used to calculate the exact effect first quantifies the propagated labels of lower level goals and then takes the (weighted) average which is then discretized again to the closest label, which is the sought label for the higher level soft goal. The quantification scale for the propagated labels may look as follows: satisficed = 2, weakly_satisficed = 1, undetermined = 0, weakly_denied = -1, denied = -2. Then, to fine-tune influences that the lower level goals from the balanced list (and thus, the propagated labels) have on the label for the higher level goal, weights can be assigned for the lower level goals in the list. Let the quantified propagated labels from the goals in the balanced list be g_i and the weights defined for each goal in the list are w_i. Then the influence of the balanced list on the higher level goal is calculated using a formula of the type: $\Sigma w_i g_i / \Sigma w_i$. The weight of a goal in a balanced list is specified by: ***has_weight_in_list***: GOAL × INTEGER × BAL_GOAL_LIST. In the following an example from the case study of a soft goal refinement by a balanced list is considered. Examples of refinement of soft goals by and-lists are provided in [7].

Table 1. The table for determining propagated labels for a higher level goal based on the satisfaction/satisficing labels of lower level goals and types of contributing links

Contributing goal label \ Type of link	satisfices	contributes_to
satisficed/satisfied	satisficed	weakly_satisficed
weakly_satisficed	weakly_satisficed	undetermined
undetermined	undetermined	undetermined
weakly_denied	weakly_denied	undetermined
denied/failed	denied	weakly_denied

Example: Consider the following set of goals and relations between them (see Fig. 1):
G7: It is required to maintain optimal number of qualified personnel
G7.1: It is required to maintain high qualification of personnel
G7.2: It is required to maintain up-to-date data on the available and needed capacity and qualifications
G7.3: It is required to maintain timely recruitment and dismissal of personnel according to the data on available and needed (human) capacity and qualifications.

Let us assume that the degrees of satisficing of the lower-level goals G7.1, G7.2 and G7.3 are known: G7.1 is satisficed, G7.2 and G7.3 are weakly satisficed. The labels are quantified and the degree of satisficing of G7.1 is considered 2 and of G7.2 and G7.3 – 1. Thus the degree of satisficing of G7 is calculated as (3*2+1+1)/5 = 1.6 which we round up to 2 (which corresponds to satisficed in our scale). Thus the label assigned to G7 is 2.

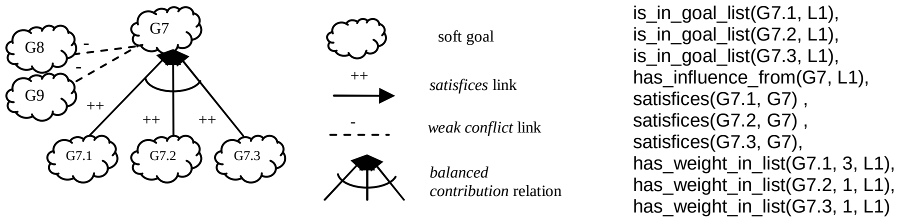

Fig. 1. The refinement of the soft goal G7 into the balanced list consisting of goals G7.1, G7.2 and G7.3; and the conflict relations between G7 and the goals G8 and G9.

When a goal is refined in one list only then the influence calculated using the described above rules defines the satisficing label of the goal. Sometimes a goal is refined into several influence lists related by *or*. This reflects the knowledge that these lists are in conflict or competition and if one is satisficed then the probability that the rest will also be satisficed is lower. In such situations we use the following strategy: first the influences of the and- and balanced lists are calculated separately and the highest among them label is assigned to the higher level goal.

Apart from the refinement links discussed so far, we can also define conflicts, which represent negative relations between goals or lists of goals.

conflicts_with: AND_GOAL_LIST × AND_GOAL_LIST: Represents joint negative effect between lists of goals – the goals in both lists cannot be satisfi(c)ed or weakly satisficed at the same time. More precisely, if all goals in one list are satisfi(c)ed then at least one goal in the other is failed or denied; if all goals in one list are at least weakly satisficed, at least one goal in the other is at most weakly denied.

weakly_conflicts_with: AND_GOAL_LIST × AND_GOAL_LIST: Represents weak joint negative effect between lists of goals, i.e., the goals in both lists cannot be satisfi(c)ed at the same time. More precisely, if all goals in one list are satisfi(c)ed then at least one goal in the other is at most weakly denied; if all goals in one list are at least weakly satisficed then at least one goal in the other is at most weakly satisficed.

Conflicts can be defined at each two levels of the goal hierarchy, however if the hierarchy is sufficiently complete then these conflicts should be propagated through the goal refinement to the lowest level at which the sources of the conflicts can be found. Conflicts can also be used at the analysis and evaluation phases for propagating satisfaction labels when only partial information is available. For example let goals g1 and g2 be in conflict at the lowest level of the goals structure and g1 is known to be satisfied. Then if the satisfaction label of g2 is not known it can be assumed at most weakly denied if g2 is soft and failed if g2 is a hard goal. If however it is known that g2 is satisfi(c)ed then there is an inconsistency in the specification. In the example on Fig.1 goals G8 “It is required to minimize training for personnel” and G9 “It is required to minimize recruitment of personnel” are in weak conflict with G7.

4 Goal-based Evaluation of Performance

Every task performed in an organization contributes to the satisfaction of a certain organizational goal(s). Tasks are realized by processes in the organization’s workflow. Each goal is formed based on a PI(s). This PI(s) can be measured (directly or indirectly) during or after the process execution depending on the goal evaluation type, in the end or during a certain period of time (an evaluation period defined as a goal horizon). Then, by comparing the measured value(s) with the goal expression(s), the satisfaction (degree of satisficing) of the goal(s) is determined. In many cases however it is not feasible (e.g., too expensive or difficult) or not possible to monitor and measure all necessary PIs. Then the mechanisms for propagating goal satisfaction values through the goals hierarchy can be used. It is only necessary to evaluate the PIs of the low-level goals. The obtained goal satisfaction values are propagated using the rules from Section 3, upwards in the goal hierarchy for determining the satisfaction (degree of satisficing) of higher-level goals. Thus, the organizational performance is evaluated by determining the satisfaction (degree of satisficing) of key organizational goals. The same principles can be applied for evaluation of agent performance.

As illustration of the proposed performance evaluation procedure consider the following example. For estimating the performance of the organization introduced in Section 1, the degree of satisficing of the high-level soft goal G3 (defined in Section 2) has to be determined. G3 is refined into an and-list of more specific goals including hard goal G3.1 (Section 2) to restrict the duration of the short-term planning process and other goals on efficiency of forward planning, plans distribution, realization of plans, etc. G3.1 is linked to G3 by a satisfices-link, therefore satisfices(G3.1, G3), G3.1 in turn is refined into an and-list consisting of two hard goals: G3.1.1 “It is required to achieve that within 48 hours from receiving a new contract, a new short-term plan is produced” and G3.1.2 “It is required to achieve that within 48 hours from receiving data about necessary changes in the short-term plan, an updated short-term plan is produced”. The PIs corresponding to G3.1.1 and G3.1.2 are PI1:“time needed to create a short term plan” and PI2:“time needed to update a short term plan”. These lower-level goals are related to tasks specified in the task graph: G3.1.1 is related to the task T1:“generate a new short-term plan” and G3.1.2 to the task T2:“update short term plan”.

By measuring the actual task execution during the evaluation period defined for G3.1 (a month), it is determined that values for both PIs corresponding to G3.1.1 and G3.1.2 (PI1 and PI2) do not exceed 48 hours. Therefore the hard goals G3.1.1 and G3.1.2 are satisfied. Due to the refinement relations, it can be concluded that G3.1 is also satisfied, therefore it contributes maximally to the satisfaction of G3 – if for example all other goals in the refinement are satisfi(c)ed then G3 is considered satisficed. G3 is at the highest level of the goals hierarchy, thus its satisfaction gives a strong positive evaluation of the overall organizational performance.

5 Methodological Issues of Goal Design

Usually, high level goals of a company are of a strategic (long-term) type. Such goals are often made operational by refining them into lower-level tactical (short-term) goals (a top-down approach). The refinement of goals may proceed until subgoals are found, which could be realized by lowest-level tasks from the task hierarchy. In practice, the top-down design approach is often combined with the bottom-up approach, performed by aggregation of goals. In the goal elicitation approach described in [4] subgoals are identified by asking "how" questions about the goals already defined, and parent goals are identified by asking "why" questions. Many of the strategic and tactical goals can be extracted from organizational documents (e.g., organizational strategy, mission statements, policies). Although, goals and objectives may not be stated very precisely in these documents, still in most cases PI expressions can be extracted from them. Further, the obtained PI expressions are used to formulate formal goals. Relations between goals can be identified via relations in the PI structures [6]. To ensure consistency of goal and PI structures, consistency checks can be performed as follows. If goals are related by refinement relation, then the PIs corresponding to these goals are related by causality relation. If the PI expressions for goals related by refinement, contain comparison functions (i.e., '>','<') or measures of degrees (i.e., 'high', 'low'), or goal patterns are specified by change functions (i.e., 'increased', 'decreased'), then the specific type of causality may be determined. For example, the goal "It is required to limit the duration of the reviewing process to a month" (with PI expression "duration of reviewing process < 1month") has a subgoal: "It is desired to increase the number of reviewers" (goal pattern "increase(number of reviewers)"). Since goals are related by refinement, the PIs "number of reviewers" and "duration of reviewing process" should be related by negative causality relation (e.g., increasing the first PI decreases the second PI).

The identification of conflict relations between goals is of importance for the design and evaluation of organizations. To identify conflicts, the goal patterns and the PIs structure can be used: by knowing the type of a causality relation between PIs and the types of goal patterns, the presence of a conflict between goals can be determined. For example, the goal "It is required to maximize the time spent on examining a plan proposal for correctness" and the goal "It is required to minimize the time spent on producing a correct plan" are in conflict, since the PIs "the time spent on examining a plan proposal for correctness" and "the time spent on producing a correct plan" are related by positive causality relation, and the corresponding goal patterns are based on opposite functions: maximize and minimize. If needed, consistency can be achieved applying conflict resolution techniques [9, 8].

6 Discussion

Goal-oriented modeling is given a special place in the area of enterprise engineering. Some aspects of our definition of a goal are inspired by state-of-the-art approaches in enterprise modeling and requirement engineering. There are however significant differences. For the first time in this framework the concept of a goal is defined based on PIs, to reflect the way the notions are used in practice and provide intuitive mechanisms for performance evaluation. Though large body of research on PIs exists in management science, PIs are hardly ever considered in enterprise modeling [6].

In the enterprise modeling framework *CIMOSA* [2] the notion of objective is used to represent business goals for a domain of the enterprise. Unlike this framework, no distinction is made between hard and soft goals. Relations between the objectives are not defined and no hierarchy of objectives is built. Also in the *TOVE framework* [2] hard and soft goals are not differentiated. Goals can be decomposed in AND/OR subgoal trees. Goals are treated very simply in the *Aris* enterprise architecture [2]: no distinction between soft and hard goals is made, also refinement of goals is not elaborated. The methodology *GRAI/GIM* [2] models performance indicators in the context of decision making, without taking into account relationships with goals.

The *i* approach* [11] focuses on the dependencies relationships between the actors. The approach models hard and soft goals and goal dependency relationships between actors w.r.t. goals. In contrast to our approach, goals in *i** are specified informally, no unified representation is enforced and no relation to PIs is established. The goal and task hierarchies are coupled in *i**, tasks are decomposed to goals and tasks. In the framework proposed, models of tasks and goals can be specified and analyzed separately. Furthermore, relations between goals and tasks can be established and used in analysis in the proposed framework. *Tropos* [1] is a methodology for agent-oriented software development based on *i** and goals are treated similarly. The *KAOS* methodology [4] focuses on requirements elaboration and provides support in connecting high-level goals to operations, objects and constraints to be implemented by the software. A goal is defined as an objective to be achieved by the system; an operational objective is called a constraint. Soft goals are not considered. Goals and constraints are defined formally using the patterns *achieve*, *cease*, *maintain*, *avoid*, *optimize*, which are reused in our approach in the notion of a goal pattern. A difference is that in our approach a goal pattern establishes a direct relation to a PI expression. KAOS goals are formalized using temporal logic and structured and operationalized to constraints in AND/OR graphs. In our approach a wider set of goals is considered, some of which cannot be expressed as temporal logic formulae. The *NFR framework* [5] focuses on representing non-functional requirements on the software system as interrelated goals. Three types of goals are defined: NFR, satisficing and argumentation goals. The last two model design decisions and arguments. The NFR goals are soft goals which can be refined using different types of relationships describing how the satisficing of the offspring relates to the satisficing of the parent goal. The label propagation procedure in our approach is inspired by but different from the one of the NFR framework. We consider only positive refinement links; negative links are modeled by conflict links. Besides, we enrich the refinement with a new balanced type of relation for soft goals, not previously used in literature, for finer definition of joint contributions of sets of goals to the satisficing of higher-level goals. Furthermore, in contrast to NFR we relate goals to PIs and other concepts (e.g., tasks, roles, agents) explicitly, which enables different types of analysis across different views on organizations.

Agent goals in multi-agent systems are specified by declarative logical specifications that describe states of the agent system, which are desirable and could be realized by the agent. Only hard goals are considered. Such declarative goals often have a simple (software-oriented) format and are often operationalized in agent programming languages by sequences of actions or plans [10]. Then, the distinction between goals and tasks, essential for our framework, is not tangible any more.

In summary, this paper presents a formal goal-oriented modeling approach in the context of the performance-oriented view on organizations. The proposed approach is based on the idea that goals should be defined over organizational performance indicators. The proposed approach includes a diverse vocabulary to express goal-related concepts and relations, in particular w.r.t. performance evaluation, e.g., organizational or individual, hard or soft goals, how they contribute to or conflict with each other's satisfaction, mechanisms for identifying the (level of) satisfaction/ satisficing of goals, as well as guidelines and techniques for building consistent goal structures. Goals are related to concepts described in other views of the framework as well, which enables different types of analysis within and between views; they are mentioned here but will be elaborated and applied on larger case studies elsewhere.

References

1. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J., 2004. Tropos: An Agent-Oriented Software Development Methodology. *Autonomous Agents and Multi-Agent Systems, 8*, 203-236.
2. Bernus, P. et al. (eds.): Handbook on Architectures of Information Systems, Springer-Verlag, Heidelberg (1998) 209-241.
3. Chan, F.T.S., 2003. Performance measurement in a supply chain *International Journal of Advanced Manufacturing Technology* 21(7), 534-548.
4. Dardenne, A., van Lamsweerde, A., Fiskas, S.: Goal Directed Requirements Acquisition. Science of Computer Programming, 20 (1993) 3-50
5. Mylopoulos, J., Chung, L., Nixon, B., 1992. Representing and Using Nonfunctional Requirements: A Process-Oriented Approach. *IEEE Transactions on Software Engineering, 18(6)*, 483-497.
6. Popova, V., Sharpanskykh, A., 2007. Modeling Organizational Performance Indicators. In *Proc. of Int. Modeling and Simulation Multiconference,* 165-170.
7. Popova, V., Sharpanskykh, A., 2008. Formal Goal-based Modeling of Organizations. *Technical Report 080103AI*, Vrije Universiteit Amsterdam, http://hdl.handle.net/1871/11732
8. Sycara, K., 1988. Resolving Goal Conflicts via Negotiation. In *Proc. of the 7th National Conference on Artificial Intelligence*, 245-250.
9. Van Lamsweerde, A., Darimont, R., Letier, E., 1998. Managing Conflicts in Goal-Driven Requirements Engineering. *IEEE Transaction on Software Engineering, 24(11)*, 908-926.
10. Van Linder, B., van der Hoek, W., and Meyer, J.-J. Ch., 1996. Formalising motivational attitudes of agents: On preferences, goals and commitments. In *Intelligent Agents Volume II*, vol. LNCS 1037, Springer, 17-32.
11. Yu, E., 1997. Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. *3rd IEEE Int. Symp. on Requirements Engineering*, 226-235.